

Design Docs

17-313 Fall 2025

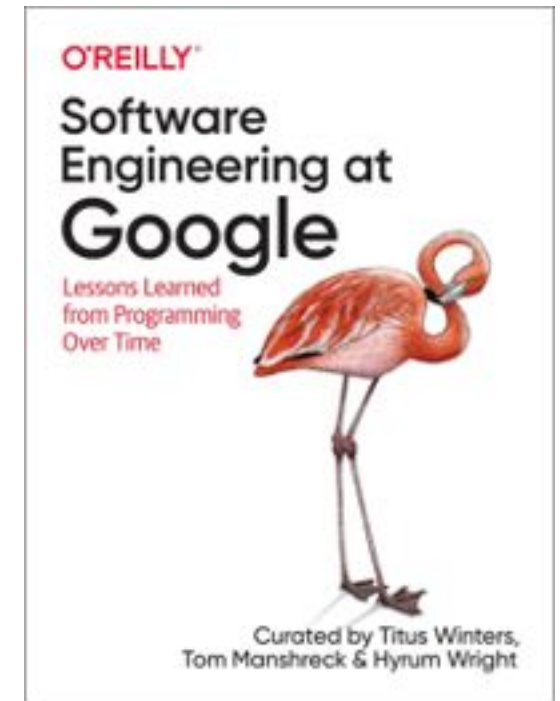
Foundations of Software Engineering

<https://cmu-17313q.github.io>

Eduardo Feo-Flushing

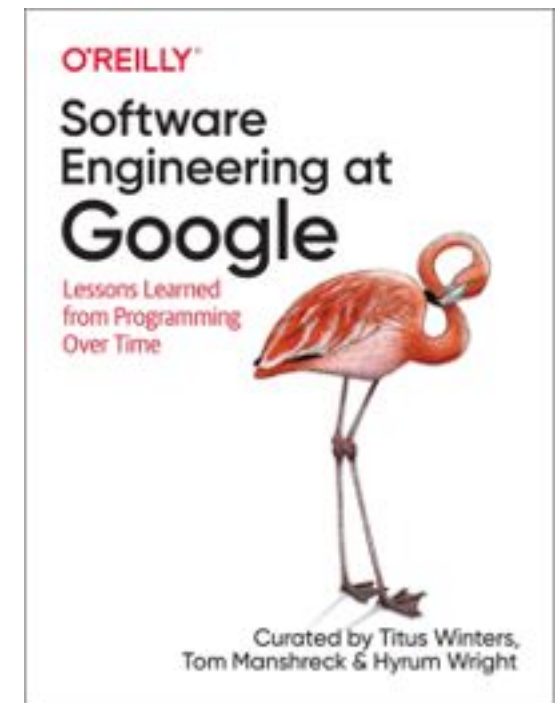
Types of documentation

- Reference documentation (incl. code comments)
- Design documents
- Tutorials
- Conceptual documentation
- Landing pages



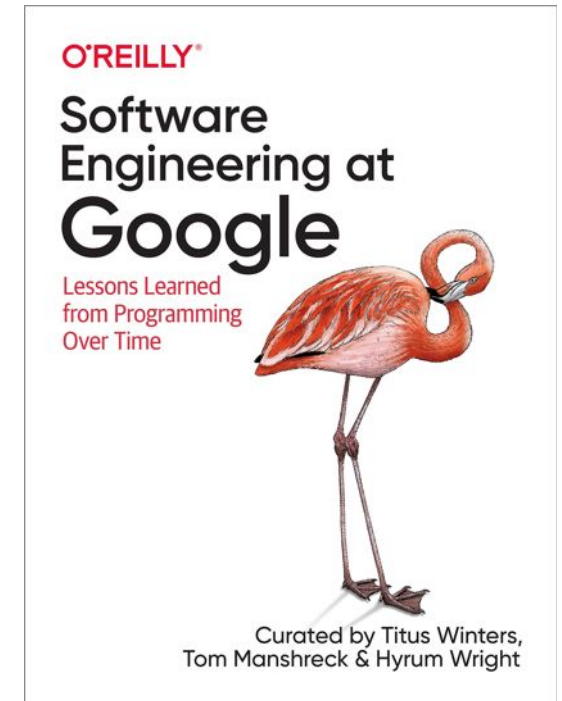
Design documents

- Code review before there is code!
- Collaborative (Google Docs)
- Ensure various concerns are covered, such as: security implications, internationalization, storage requirements, and privacy concerns.
- A good design doc should cover
 - Goals and use cases for the design
 - Implementation ideas (not too specific!)
 - Propose key design decisions with an emphasis on their individual tradeoffs



Design documents

- The *best* design docs suggest design goals, and cover alternative designs, documenting the strengths and weaknesses of each.
- The *worst* design docs accidentally embed ambiguities, which cause implementers to develop contradictory solutions that the customer doesn't want.



Companies using an RFC-like engineering planning process*

<ul style="list-style-type: none"> • Airbnb • Affirm • Algolia • Amazon • AutoScout24 • Asana • Atlassian • Blue Apron • Bitrise • Booking.com • Brex • BrowserStack • Canonical • Carousell • Catawiki • Cazoo • Cisco • CockroachDB • Coinbase • Comcast Cable • Container Solutions • Contentful • Couchbase • Criteo • Curve • Daimler • Delivery Hero 	<ul style="list-style-type: none"> • Doctolib • DoorDash • Dune Analytics • eBay • Ecosia • Elastic • Expedia • Glovo • Gojek • Grab • Faire • Flexport • GitHub • GitLab • GoodNotes • Google • Grafana Labs • GrubHub • HashiCorp • Hopin • Hudl • Indeed • Intercom • LinkedIn • Kiwi.com • Klarna • MasterCard 	<ul style="list-style-type: none"> • Mews • MongoDB • Monzo • Mollie • Miro • N26 • Netlify • Nobl9 • Notion • Nubank • Oscar Health • Octopus Deploy • OLX • Onfido • Pave • Peloton • Picnic • PlanGrid • Preply • Razorpay • Reddit • Red Hat • SAP • Salesforce • Shopify • Siemens • Spotify • Square 	<ul style="list-style-type: none"> • Stripe • Synopsys • Skyscanner • SoundCloud • Sourcegraph • Spotify • Stedi • Stream • SumUp • Thumbtack • TomTom • Trainline • TrueBill • Trustpilot • Twitter • Uber • VanMoof • Virta Health • VMWare • Wayfair • Wave • Wise • WarnerMedia & HBO • Zalando • Zapier • Zendesk • Zillow
---	---	--	--

*not a complete list

pragmaticengineer.com

Common parts/templates

1. Metadata: *version, date, authors*
2. Executive Summary: *problem being solved, project mission*
3. Stakeholders (and non-stakeholders)
4. Scenarios / User Stories
5. User Experience

```
/*  
 * Dear maintainer:  
 *  
 * Once you are done trying to 'optimize' this module  
 * and have realized what a terrible mistake that was,  
 * please increment the following counter as a warning  
 * to the next person:  
 *  
 * total hours wasted here = 342  
 *  
 */
```

4. Non-Goals
5. Roadmap / Timeline
6. Open Issues

Examples: SourceGraph RFCs

Requests for Comment



Meet Cody 

We're building the only AI coding assistant that knows your **entire codebase**

Cody answers technical questions and writes code directly in your IDE, using your code graph for context and accuracy.

Get Started with Cody **BETA**



GitHub



GitLab



Google

By registering, you agree to our [Terms of Service](#) and [Privacy Policy](#).

When to use an RFC:



- You want to frame a problem and propose a solution.
- You want thoughtful feedback from team members on our globally-distributed remote team.
- You want to surface an idea, tension, or feedback.
- You want to define a project or design brief to drive project collaboration.
- You need to surface and communicate around a highly cross-functional decision with our [formal decision-making process](#).

Don't use an RFC when



- You want to discuss personal or sensitive topics one-on-one with another team member.
- You want to make a decision to change something where you are the decider. In the vast majority of cases, creating an RFC to explain yourself will be overkill. RFCs should only be used if a decision explicitly requires one of the bullets in the previous page.

RFC Labels



- **WIP:** The author is still drafting the RFC and it's not ready for review.
- **Review:** The Review label is used when the RFC is ready for comments and feedback.
- **Approved:** When the RFC is for the purpose of making a decision, the Approved label indicates that the decision has been made.
- **Implemented:** When the RFC is for the purpose of making a decision, the Implemented label indicates that the RFC's proposal has been implemented.
- **Closed:** When the RFC is for the purpose of collaboration or discussion but not necessarily to make a decision or propose a specific outcome that will eventually become Implemented, the Closed label indicates that the RFC is no longer an active collaborative artifact.
- **Abandoned:** When the RFC is for the purpose of making a decision, and there are no plans to move forward with the RFC's proposal, the Abandoned label indicates that the RFC has been purposefully set aside.

Observe Sourcegraph Design Docs

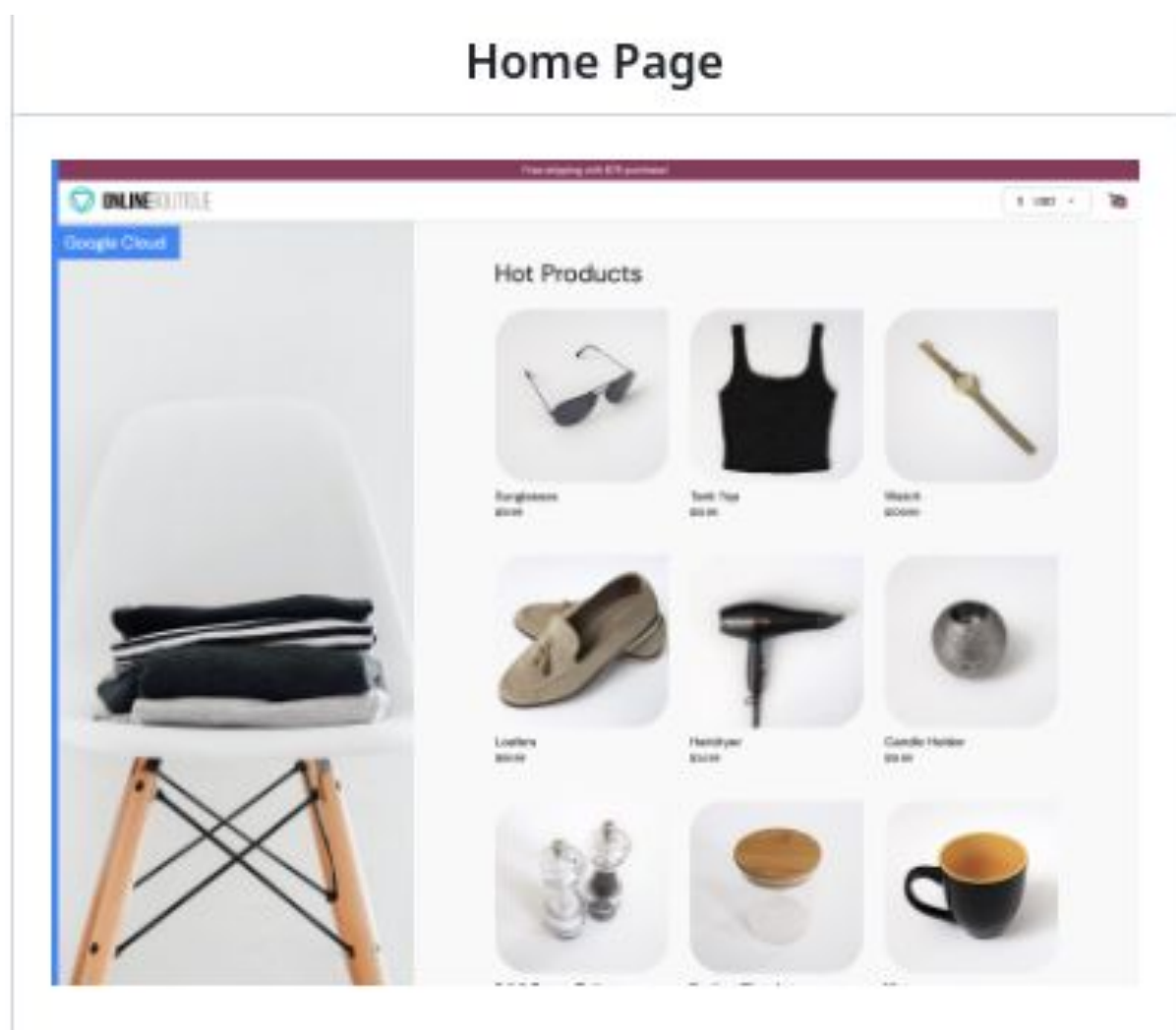
- Docs are publicly available

<https://drive.google.com/drive/folders/1zP3FxdDlcSQGC1qvM9IHZRHH4I9jwwa>

- Let's take a look at one!

Exercise

- 3 Proposed Features:
 - Add Payment Method
 - Personalized Promotions
 - Add Android Support



Try the demo: bit.ly/313-boutique